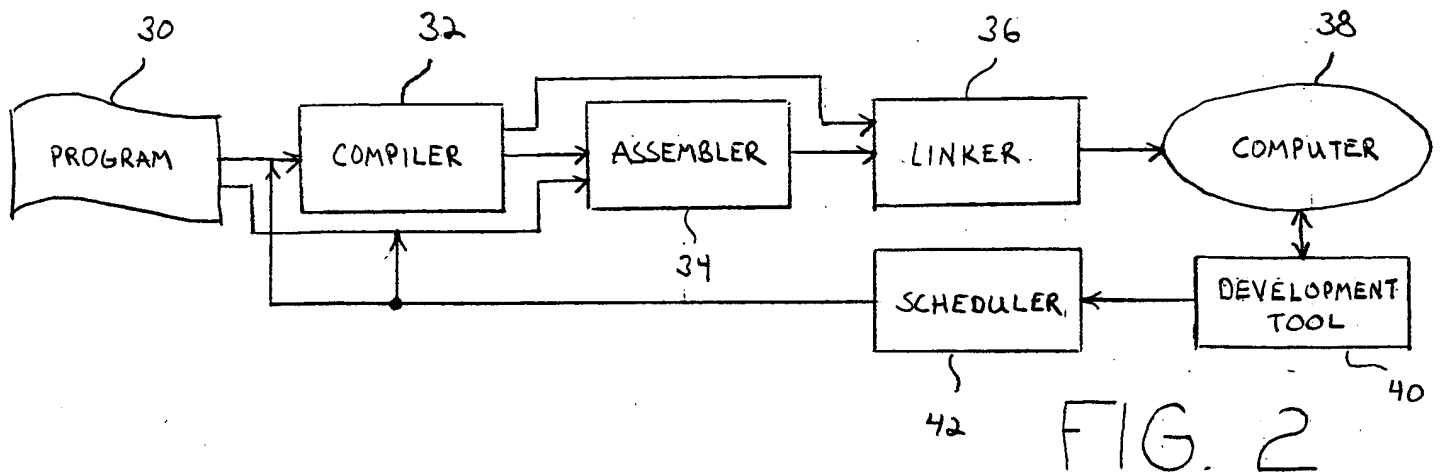


FIG. 1



$*bp = (*bp \& rm[offset]) \mid ((cd \ll offset) \& mask)$  ← 50

1:	const	tmp1,mask	;load mask
2:	consth	tmp1,mask	
3:	sll	tmp2,cd,offset	;cd<<offset
4:	and	tmp2,tmp1,tmp2	; (cd<<offset) & mask
5:	const	tmp1,_rm	;base address rm
6:	consth	tmp1,_rm	
7:	add	tmp1,tmp1,offset	;address of rm[offset]
8:	load	tmp1,(tmp1)	;rm[offset]
9:	load	bp,(&bp)	;*bp
10:	and	tmp1,bp,tmp1	;*bp & rm[offset]
11:	or	tmp1,tmp1,tmp2	;final expression
12:	store	tmp1,(&bp)	;assign *bp

52

Decode				Execute					Writeback		Cycle
I0	I1	I2	I3	ALU1	ALU2	SHF	LS	BRN	R1	R2	
1:const	2:consth	3:sll	4:and	1:const		3:sll					1
5:const	6:consth	7:add	8:load		2:consth				1	3	2
9:load	10:and	11:or	12:store	4:and	5:const				2		3
				6:consth					4	5	4
					7:add				6		5
							8:load		7		6
							9:load				7
									8		8
				10:and					9		9
					11:or				10		10
						12:store			11		11
											12

54

FIG. 3

Stage	Issue 1	Issue 2	Issue 3	Issue 4
Write Back	none	none	none	none
Execute	0x1000	0x1001	0x1002	0x1003
Memory 1	0x1006	0x1007	0x1008	none
Memory 0	0x100A	none	none	none
Address Generation	0x100B	0x100D	0x100F	0x1011
Read Data	none	none	none	none
Grouping	0x1013	0x1014	0x1015	0x1016
Fetch/Decode	0x1017	0x1018	0x1019	none

← 60

FIG 4

70 72

11 Cycles		TARGET DISASSEMBLED CODE	
--	0xffff	:	nop
--	0x1000	EX	_start:: bits %fmode, 0x2
--	0x1001	EX	:
--	0x1002	EX	:
--	0x1003	EX	:
--	0x1006		:
--	0x1007		:
--	0x1008		:
--	0x100a		:
--	0x100b		:
--	0x100d		:
--	0x100f		:
--	0x1011		:
--	0x1013	GR	lp_calc:: ldu r4, r12, +1
--	0x1014	GR	:
--	0x1015	GR	:
--	0x1016	GR	:
--	0x1017	FD	:
--	0x1018	FD	:

← 64

↑  
66

↑  
62

↑  
68

↑  
74

↑  
76

FIG. 5

Disassembly

469 cycles

TARGET DISASSEMBLED CODE

```

-- 0x1e0      GR      :      ldd    r2, a7, 0x1
-- 0x1e1      GR      :      mov    a6, a7
-- 0x1e2      GR      :      add    a6, 3
-- 0x1e3      GR      :      ldd    r4, a6
-- 0x1e4      FD      :      mov.e   a1, r2
-- 0x1e5      NEFD     :      mov.e   a0, r4
-- 0x1e6      NEFD     :      ld      r6, a0
-- 0x1e7      NEFD     :      st      r6, a1
-- 0x1e8      NEFD     :      ld      r6, a1
-- 0x1e9      NEFD     :      iadd.e  r4, 1
-- 0x1ea      NEFD     :      std     r4, a6
-- 0x1ec      NEFD     :      iadd.e  r2, 1
-- 0x1ed      NEFD     :      std     r2, a7, 0x1
-- 0x1ef      RD      :      cmp     r6, 0
-- 0x1f0      RD      :      bnz     0x1e1
-- 0x1f1      NEFD     :      ldd    a0, a7, 0x5
-- 0x1f2      NEFD     :      add    a7, 6
-- 0x1f3      NEFD     :
-- 0x1f4      FUNC_EXIT strcpy::  ret
-- 0x1f5      write sdsp::  pushd   r10, a7
  
```

80

FIG. 6

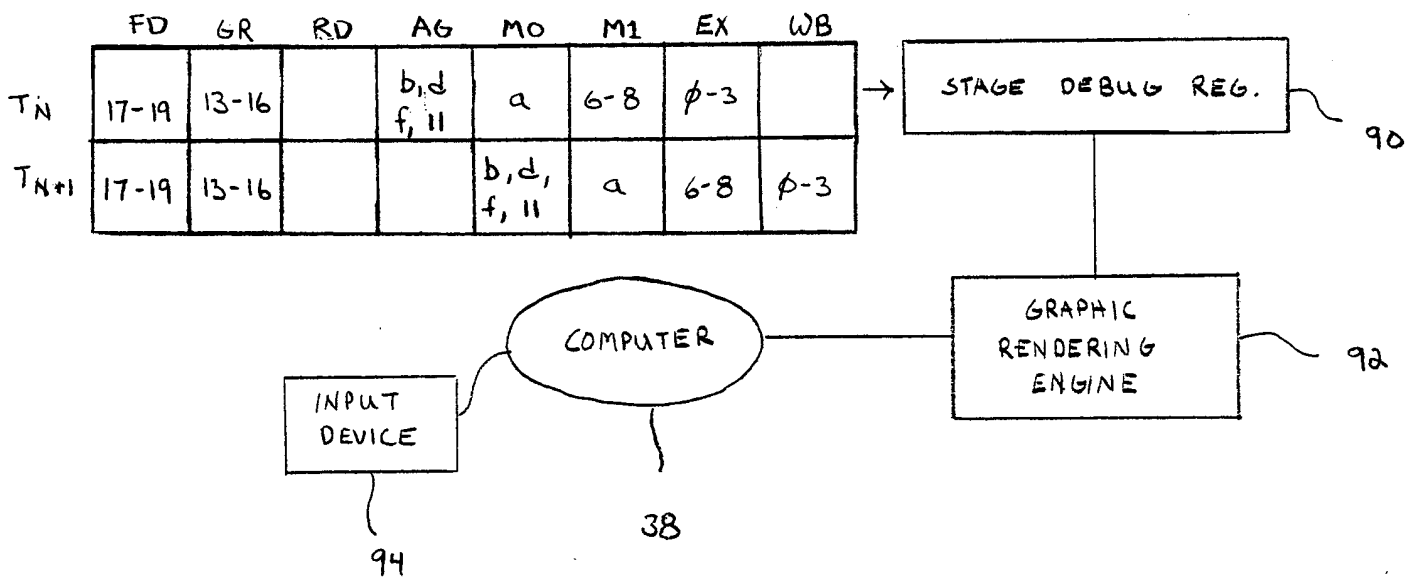


FIG. 7